

AI Client APIs Documentation

This document describes the APIs available for interacting with various AI providers (Anthropic, Google Gemini, and OpenAI).

Common Interface

All providers implement the following interface, and these are what you use to make queries to the provider. See `pkg/api/client.go` for the full interface definition.

```
// pkg/api/client.go

const (
    Anthropic Provider = "anthropic"
    Gemini    Provider = "gemini"
    OpenAI    Provider = "openai"
)

func NewClient(provider Provider) (Client, error)

type Client interface {
    QueryText(prompt string, model string, options Options) (string, error)
    QueryJSON(prompt string, model string, options Options) (string, error)
}
```

Usage Example

```
// Create a new client
client, err := NewClient(OpenAI)
if err != nil {
    log.Fatal(err)
}

// Query for text
response, err := client.QueryText("Tell me a joke", "gpt-4-turbo-preview",
Options{})
if err != nil {
    log.Fatal(err)
}

// Query for JSON
jsonResponse, err := client.QueryJSON("Return a JSON object with today's
weather", "gpt-4-turbo-preview", Options{})
if err != nil {
    log.Fatal(err)
}
```

Error Handling

All methods return errors in the following cases:

- Missing API keys
- Empty or invalid prompts
- API request failures
- Invalid responses
- JSON parsing errors (for QueryJSON)

Environment Variables

The following environment variables must be set:

- `ANTHROPIC_API_KEY` - For Anthropic Claude API access
- `GEMINI_API_KEY` - For Google Gemini API access
- `OPENAI_API_KEY` - For OpenAI API access

Provider-Specific Implementations

Anthropic Client

The Anthropic client uses Claude models for text and JSON generation.

Models

- Tested with: `claude-3-sonnet-20240229`

Methods

QueryText

- Sends a text prompt to Claude and returns a natural language response
- Maximum token limit: 2048
- Returns error if prompt is empty

QueryJSON

- Generates structured JSON output from a prompt
- Returns error if response is not valid JSON
- Returns error if prompt is empty

Google Gemini Client

The Gemini client interfaces with Google's Gemini models.

Models

- Tested with: `gemini-2.0-flash-exp`

Methods

QueryText

- Generates text content with MIME type "text/plain"
- Concatenates multiple response parts if present
- Requires GEMINI_API_KEY environment variable

QueryJSON

- Generates JSON content with MIME type "application/json"
- Returns structured data in JSON format
- Requires GEMINI_API_KEY environment variable

OpenAI Client

The OpenAI client interfaces with GPT models via the OpenAI API.

Models

- Tested with: `gpt-4-turbo-preview`

Methods

QueryText

- Sends chat completion requests to OpenAI
- Default max tokens: 1024
- Requires OPENAI_API_KEY environment variable
- Returns error if API key not set

QueryJSON

- Generates structured JSON responses
- Default max tokens: 1024
- Returns error if prompt is empty
- Requires OPENAI_API_KEY environment variable